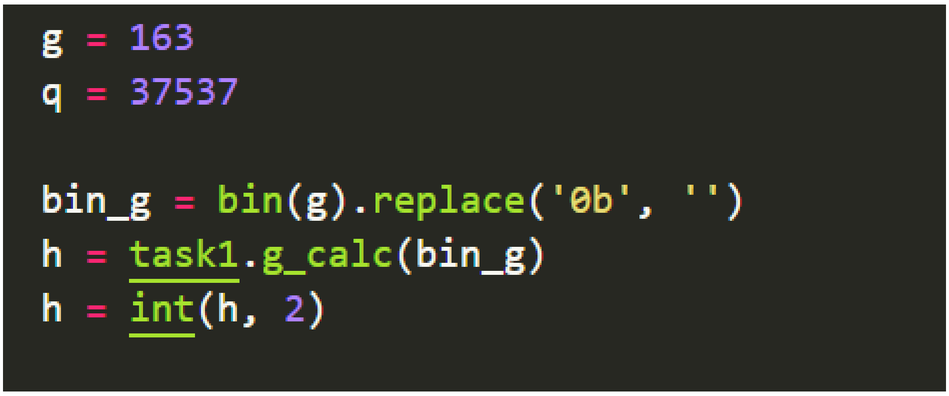
Task 6 Fixed Length Hash



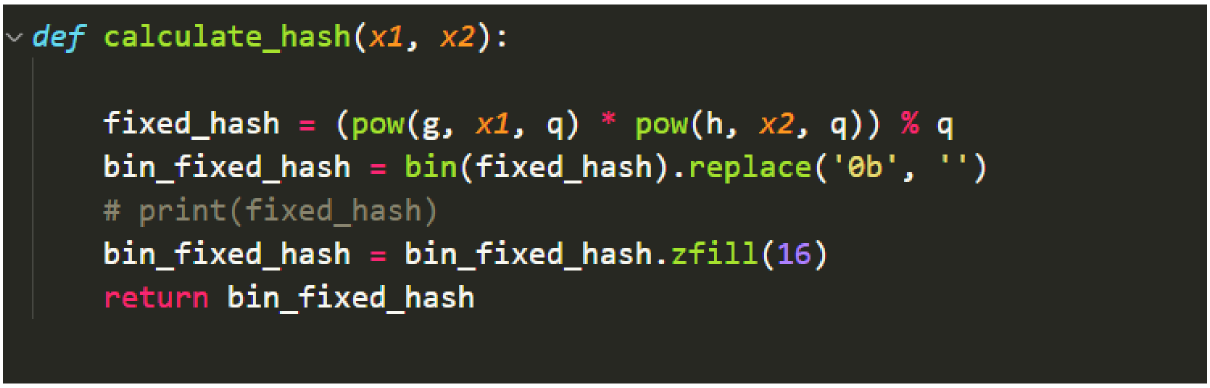
```
g = 163
q = 37537

bin_g = bin(g).replace('0b', '')
h = task1.g_calc(bin_g)
h = int(h, 2)
```

G = Generator of the cyclic group 37537

Q = Order of the group

H = random number generated from the given group.



```
def calculate_hash(x1, x2):

    fixed_hash = (pow(g, x1, q) * pow(h, x2, q)) % q
    bin_fixed_hash = bin(fixed_hash).replace('0b', '')
    # print(fixed_hash)
    bin_fixed_hash = bin_fixed_hash.zfill(16)
    return bin_fixed_hash
```

Calculate_hash()

The function takes, x1 and x2 as input parameters from the user and calculates g^x1 * h^x2 and mods with q. The resultant binary hash gives us on value g^x1 of size 16 bits and h^x2 of 16 bits, a result of 16 bits is generated halving the input size.

Sample Input

X1: 12231

X2: 37523

Sample Output

Output: 0011000100101110 16